

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-295882

(43)Date of publication of application : 10.11.1995

(51)Int.Cl.

G06F 12/08

(21)Application number : 06-084321

(71)Applicant : HITACHI LTD

(22)Date of filing : 22.04.1994

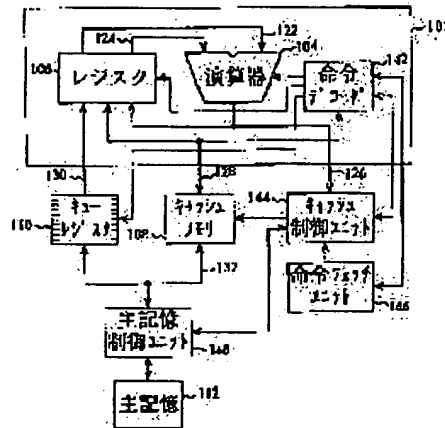
(72)Inventor : SHIMAMURA KOTARO
HOTTA TAKASHI
TANAKA SHIGEYA
YAMAGATA MAKOTO

(54) INFORMATION PROCESSOR AND INFORMATION PROCESSING SYSTEM

(57)Abstract:

PURPOSE: To attain high-speed processing by executing a preload instruction while leaving enough time to spare before data is processed, and reading the data out of a queue register and processing it at the time of processing the data actually.

CONSTITUTION: An instruction processing unit 102, an instruction decoder 142, a computing element 104, a register 106, a cache control unit 144, an instruction fetch unit 146, a cache memory 108, the queue register 110, a main storage control unit 148, a main storage device 112, data buses 122, 124, 126, 128, 130, 132 are provided. Then, the preload instruction to transfer the data to the queue register 110 from the main storage device 112 is executed while leaving enough time to spare before the data comes to be needed, and at the time of using the data, a queue transfer instruction to transfer the data from the queue register 110 to the register 106 is executed. Accordingly, the processing can be executed at high speed without being affected by the read-out time of the low-speed main storage device 112.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-295882

(43) 公開日 平成7年(1995)11月10日

(51) Int.Cl.⁶
G 0 6 F 12/08

識別記号 庁内整理番号
D 7608-5B
C 7608-5B

F I

技術表示箇所

審査請求 未請求 請求項の数32 OL (全 21 頁)

(21) 出願番号 特願平6-84321

(22) 出願日 平成6年(1994)4月22日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 島村 光太郎

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72) 発明者 堀田 多加志

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(72) 発明者 田中 成弥

茨城県日立市大みか町七丁目1番1号 株

式会社日立製作所日立研究所内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

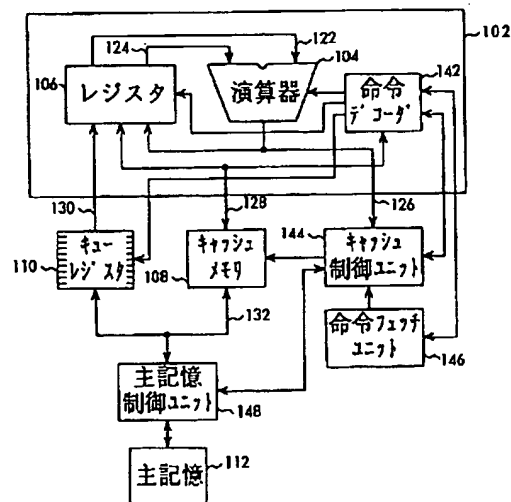
(54) 【発明の名称】 情報処理装置、及び、情報処理システム

(57) 【要約】

【目的】 情報処理装置において、キャッシュメモリに入り切らない大量のデータを処理する場合に、キャッシュミスによるペナルティの発生を防止し、性能を向上すること。

【構成】 先入れ先出し方式のレジスタ（キューレジスタ）を設け、主記憶部からキューレジスタへデータを転送するプリロード命令と、キューレジスタからレジスタへデータを転送するキュー転送命令をサポートする。データを使用する十分前にプリロード命令を実行してキューレジスタにデータを転送しておくことにより、主記憶読み出しによるペナルティなしに高速に処理を行う。

図 1



【特許請求の範囲】

【請求項1】データを格納する主記憶部と、
前記主記憶部に格納されているデータの一部を格納するキャッシュメモリと、
前記キャッシュメモリに格納されたデータを、命令に従って処理する命令処理ユニットと、
前記命令処理ユニットに接続され、この命令処理ユニットの命令の実行により前記主記憶部から読みだされるデータを格納する先入れ先出し型のキューレジスタ手段とを有する情報処理システム。

【請求項2】請求項1項の情報処理システムにおいて、前記命令処理ユニットはデータを格納するレジスタと、このレジスタに格納されたデータに対して、命令に従って所定の演算を行う演算器とを有することを特徴とする情報処理システム。

【請求項3】請求項2項の情報処理システムにおいて、前記キューレジスタ手段に格納されたデータを、命令に従って前記レジスタに格納する手段を有することを特徴とする情報処理システム。

【請求項4】請求項2項の情報処理システムにおいて、前記キューレジスタ手段に格納されたデータを命令に従って前記演算器に入力する手段を有し、前記キューレジスタ手段に格納されたデータと、前記レジスタに格納されたデータとの間で所定演算が実行されることを特徴とする情報処理システム。

【請求項5】請求項2項の情報処理システムにおいて、前記キューレジスタ手段に格納されたデータを命令に従って前記演算器に入力する手段を有し、前記キューレジスタ手段に格納されたデータ間で所定演算が実行されることを特徴とする情報処理システム。

【請求項6】請求項5項の情報処理システムにおいて、前記キューレジスタ手段は複数の先入れ先出し型のキューレジスタを有し、それぞれのキューレジスタに格納されたデータ間で所定演算が実行されることを特徴とする情報処理システム。

【請求項7】請求項1項の情報処理システムにおいて、前記命令処理ユニットは複数の演算器と、これらの演算器のそれぞれに対応して備えられた複数のレジスタとを有することを特徴とする情報処理システム。

【請求項8】請求項7項の情報処理システムにおいて、前記キューレジスタ手段は前記複数のレジスタのそれぞれに対して備えられた複数の先入れ先出し型のキューレジスタを有することを特徴とする情報処理システム。

【請求項9】請求項8項の情報処理システムにおいて、前記キューレジスタのそれぞれに格納されたデータを、命令に従って対応する前記レジスタに格納する手段を有することを特徴とする情報処理システム。

【請求項10】請求項9項の情報処理システムにおいて、前記複数の演算器は整数演算器、及び、浮動小数点演算器を含むことを特徴とする情報処理システム。

【請求項11】請求項1項の情報処理システムにおいて、前記主記憶部から前記キューレジスタ手段に書き込まれるデータの書き込み位置を、そのデータが前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段を有することを特徴とする情報処理システム。

【請求項12】請求項11項の情報処理システムにおいて、前記書き込み制御手段は前記キューレジスタ手段に書き込まれたデータの最上位の位置、及び、最後尾の位置を記憶する回路を有することを特徴とする情報処理システム。

【請求項13】請求項1項の情報処理システムにおいて、前記主記憶部が複数のバンクからなり、前記複数のバンクのそれぞれに格納されたデータを、命令実行の順序に従って前記キューレジスタ手段に書き込む順序を制御する書き込み制御手段を有することを特徴とする情報処理システム。

【請求項14】請求項13項の情報処理システムにおいて、前記書き込み制御手段は、前記各バンクに対応する複数のデータバッファと、

前記複数のバンクから前記複数のデータバッファへのデータの書き込みを制御する書き込み制御回路と、

前記キューレジスタ手段へ転送すべきデータが格納されている前記主記憶のバンクの番号を、命令実行の順序に従って記憶するバンク順序記憶回路と、

前記複数のデータバッファからのデータを、前記バンク順序記憶回路に記憶されているバンクの番号に従って選択し、前記キューレジスタ手段に書き込むための選択回路とを有する情報処理システム。

【請求項15】請求項1項の情報処理システムにおいて、前記キューレジスタ手段は前記命令処理ユニットに接続される先入れ先出し型の第1のキューレジスタと、この第1のキューレジスタと前記主記憶部との間に接続される先入れ先出し型の第2のキューレジスタとから成ることを特徴とする情報処理システム。

【請求項16】請求項15項の情報処理システムにおいて、前記主記憶部から前記第2のキューレジスタに書き込まれるデータの書き込み位置を、そのデータが前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段とを有することを特徴とする情報処理システム。

【請求項17】主記憶に格納されているデータ、及び、命令の一部を格納するキャッシュメモリ手段と、前記キャッシュメモリ手段に格納された命令の実行を行う命令処理ユニットであって、前記キャッシュメモリ手段に格納された命令をデコードする命令デコード部と、

データを格納する少なくとも1つのレジスタと、前記レジスタに格納されたデータに対して、所定の演算を行う少なくとも1つの演算器と、を有する命令処理ユ

ニットと、

前記命令処理ユニットに接続され、この命令処理ユニットの命令の実行により前記主記憶部から読みだされるデータを格納する先入れ先出し型のキューレジスタ手段とを有する情報処理装置。

【請求項18】請求項17項の情報処理装置において、前記キャッシュメモリ手段に格納される命令は、前記主記憶部から前記キューレジスタ手段へデータを転送するプリロード命令を含むことを特徴とする情報処理装置。

【請求項19】請求項17項記載の情報処理装置において、前記キャッシュメモリ手段に格納される命令は、前記キューレジスタ手段から前記レジスタへデータを転送する命令を含むことを特徴とする情報処理装置。

【請求項20】請求項18項記載の情報処理装置において、前記キャッシュメモリ手段に格納される前記プリロード命令は、前記主記憶部から前記キューレジスタ手段へのデータの転送と前記キューレジスタ手段から前記レジスタへのデータの転送とを行う複合命令を含むことを特徴とする情報処理装置。

【請求項21】請求項17項記載の情報処理装置において、前記キャッシュメモリ手段に格納される命令は、前記キューレジスタ手段からデータを削除する命令を含むことを特徴とする情報処理装置。

【請求項22】請求項17項記載の情報処理装置において、前記キャッシュメモリ手段に格納される命令は、前記キューレジスタ手段に格納されたデータを用いて前記演算器で演算を行うキュー演算命令を含むことを特徴とする情報処理装置。

【請求項23】請求項22項記載の情報処理装置において、前記キャッシュメモリ手段に格納されるキュー演算命令は、前記キューレジスタ手段に格納された複数のデータの間の演算を行う命令を含むことを特徴とする情報処理装置。

【請求項24】請求項22項記載の情報処理装置において、前記キャッシュメモリ手段に格納されるキュー演算命令は、演算の実行後に演算に使用したデータをキューレジスタから削除する命令、又は、削除しない命令を含むことを特徴とする情報処理装置。

【請求項25】請求項17項記載の情報処理装置において、前記キューレジスタ手段は、前記主記憶部に接続される複数のキューレジスタと、前記複数のキューレジスタと前記演算器を接続する手段とを有することを特徴とする情報処理装置。

【請求項26】請求項25項記載の情報処理装置において、前記命令処理ユニットが有する複数の演算器は整数演算器、及び、浮動小数点演算器を含み、これらの演算器のそれぞれに前記複数のキューレジスタが接続されていることを特徴とする情報処理装置。

【請求項27】請求項17項の情報処理装置において、前記主記憶部から前記キューレジスタ手段に書き込まれ

るデータの書き込み位置を、そのデータが前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段を有することを特徴とする情報処理装置。

【請求項28】請求項27項の情報処理装置において、前記書き込み制御手段は前記キューレジスタ手段に書き込まれたデータの最上位の位置、及び、最後尾の位置を記憶する回路を有することを特徴とする情報処理装置。

【請求項29】請求項17項の情報処理装置において、前記主記憶部が複数のバンクからなり、前記複数のバンクのそれぞれに格納されたデータを、命令実行の順序に従って前記キューレジスタ手段に書き込む順序を制御する書き込み制御手段を有することを特徴とする情報処理装置。

【請求項30】請求項29項の情報処理装置において、前記書き込み制御手段は、前記各バンクに対応する複数のデータバッファと、

前記複数のバンクから前記複数のデータバッファへのデータの書き込みを制御する書き込み制御回路と、

前記キューレジスタ手段へ転送すべきデータが格納されている前記主記憶のバンクの番号を、命令実行の順序に従って記憶するバンク順序記憶回路と、

前記複数のデータバッファからのデータを、前記バンク順序記憶回路に記憶されているバンクの番号に従って選択し、前記キューレジスタ手段に書き込むための選択回路とを有する情報処理装置。

【請求項31】請求項17項の情報処理装置において、前記キューレジスタ手段は前記命令処理ユニットに接続される先入れ先出し型の第1のキューレジスタと、この第1のキューレジスタと前記主記憶部との間に接続される先入れ先出し型の第2のキューレジスタとから成ることを特徴とする情報処理装置。

【請求項32】請求項31項の情報処理装置において、前記主記憶部から前記第2のキューレジスタに書き込まれるデータの書き込み位置を、そのデータが前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段とを有することを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、情報処理装置、及び、情報処理システムに係り、特に、離散アドレスアクセス時のデータ転送効率を改善する情報処理装置、及び、情報処理システムに関する。

【0002】

【従来の技術】従来の情報処理装置では、低速な主記憶へのアクセスによる性能低下を防ぐために、少容量ではあるが高速なキャッシュメモリを使用して性能を高める方法が用いられている。キャッシュメモリを使用して性能を高める方法については、ヘネシ&バターソン著「コンピュータ・アーキテクチャ設計・実現・評価の定量的アプローチ」(日経BP社) p.403~428に説明

されている。以下、キャッシュメモリを用いた従来の情報処理装置の動作を説明する。命令処理ユニットは、キャッシュメモリからデータを受け取り、必要な処理を行った後、結果をキャッシュメモリに転送する。命令処理ユニットの必要としているデータがキャッシュメモリに存在する場合には、キャッシュメモリに格納されているデータが命令処理ユニットに転送される。命令処理ユニットの必要としているデータがキャッシュメモリに存在しない場合には、必要なデータが主記憶からキャッシュメモリに転送され、キャッシュメモリに書き込まれる。10 それと同時に、主記憶から転送されてきたデータが命令処理ユニットに転送される。キャッシュメモリは主記憶に比べて高速であり、必要なデータがキャッシュメモリに存在する場合には、主記憶からデータを転送する場合に比べ高速に処理を行うことが可能となる。通常の処理を行う場合には、同じデータを何度も繰り返し使用することが多く、必要なデータがキャッシュメモリに存在する可能性が高いため、キャッシュメモリを設けることにより高い性能が得られる。

【0003】

【発明が解決しようとする課題】しかしながら、従来の情報処理装置では、同じデータを繰り返し使用することの少ない処理を行う場合には、キャッシュメモリに必要なデータが存在する可能性が低いため、主記憶からのデータ転送が頻繁に発生し、性能が低下する。また、キャッシュメモリの容量より大きいデータを一度に処理する場合には、あるデータをキャッシュメモリに書き込む時に別のデータをキャッシュメモリから消去する必要が生じ、キャッシュメモリに必要なデータが存在する可能性が低下するため、性能が低下する。

【0004】本発明の目的は、キャッシュメモリに必要なデータが存在する可能性の低い処理を行う場合にも、高速に処理を行うことの可能な情報処理装置、及び、情報処理システムを提供することにある。

【0005】本発明の他の目的は、離散アドレスアクセス時のデータの転送効率を改善し、高性能な情報処理装置、及び、情報処理システムを提供することにある。

【0006】更に、本発明の他の目的は、キャッシュミスによる処理効率低下を改善し、高性能な情報処理装置、及び、情報処理システムを提供することにある。

【0007】

【課題を解決するための手段】本発明の情報処理システムによれば、データを格納する主記憶部と、前記主記憶部に格納されているデータの一部を格納するキャッシュメモリと、前記キャッシュメモリに格納されたデータを、命令に従って処理する命令処理ユニットと、前記命令処理ユニットに接続され、この命令処理ユニットの命令の実行により前記主記憶部から読みだされるデータを格納する先入れ先出し型のキューレジスタ手段とを有する。

【0008】前記命令処理ユニットは通常、データを格納するレジスタと、このレジスタに格納されたデータに対して、命令に従って所定の演算を行う演算器とを有する。本発明の実施態様によれば、前記キューレジスタ手段に格納されたデータを、命令に従って前記レジスタに格納する手段を有する。

【0009】本発明の他の実施態様によれば、前記キューレジスタ手段に格納されたデータを命令に従って前記演算器に入力する手段を有し、前記キューレジスタ手段に格納されたデータと、前記レジスタに格納されたデータとの間で所定演算が実行される。

【0010】本発明の他の実施態様によれば、前記キューレジスタに格納されたデータを命令に従って前記演算器に入力する手段を有し、前記キューレジスタ手段に格納されたデータ間で所定演算が実行される。

【0011】本発明の他の実施態様によれば、前記キューレジスタ手段は複数の先入れ先出し型のキューレジスタを有し、それぞれのキューレジスタに格納されたデータ間で所定演算が実行される。

20 【0012】本発明の他の実施態様によれば、前記主記憶部から前記キューレジスタ手段に書き込まれるデータの書き込み位置を、そのデータが前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段を有する。

【0013】前記書き込み制御手段は前記キューレジスタ手段に書き込まれたデータの最上位の位置、及び、最後尾の位置を記憶する回路を有することが望ましい。

30 【0014】本発明の他の実施態様によれば、前記主記憶部が複数のバンクからなり、前記複数のバンクのそれぞれに格納されたデータを、命令実行の順序に従って前記キューレジスタ手段に書き込む順序を制御する書き込み制御手段を有する。

【0015】前記書き込み制御手段は、前記各バンクに対応する複数のデータバッファと、前記複数のバンクから前記複数のデータバッファへのデータの書き込みを制御する書き込み制御回路と、前記キューレジスタ手段へ転送すべきデータが格納されている前記主記憶部のバンクの番号を、命令実行の順序に従って記憶するバンク順序記憶回路と、前記複数のデータバッファからのデータを、前記バンク順序記憶回路に記憶されているバンクの番号に従って選択し、前記キューレジスタ手段に書き込むための選択回路とを有することが望ましい。

40 【0016】本発明の情報処理装置によれば、主記憶部に格納されているデータ、及び、命令の一部を格納するキャッシュメモリ手段と、キャッシュメモリ手段に格納された命令をデコードする命令デコード部と、データを格納する少なくとも1つのレジスタと、前記レジスタに格納されたデータに対して、所定の演算を行う少なくとも1つの演算器とを有する命令処理ユニットと、前記命令処理ユニットに接続され、この命令処理ユニットの命令

の実行により前記主記憶部から読みだされるデータを格納する先入れ先出し型のキューレジスタ手段とを有する。

【0017】前記キャッシュメモリ手段に格納される命令は、前記主記憶部から前記キューレジスタ手段へデータを転送するプリロード命令を含む。

【0018】本発明の実施態様によれば、前記キャッシュメモリ手段に格納される命令は以下の命令を含む。

【0019】(1) キューレジスタ手段から前記レジスタへデータを転送する命令

(2) 主記憶部からキューレジスタ手段へのデータの転送とキューレジスタ手段から前記レジスタへのデータの転送とを行う複合命令

(3) キューレジスタ手段からデータを削除する命令

(4) キューレジスタ手段に格納されたデータを用いて前記演算器で演算を行うキュー演算命令

(5) キューレジスタ手段に格納された複数のデータの間の演算を行う複数データキュー演算命令

(6) 演算実行後に演算に使用したデータをキューレジスタ手段から削除するキュー演算削除命令、又は、削除しないキュー演算非削除命令を含むキュー演算命令

本発明の他の実施態様によれば、前記主記憶部から前記キューレジスタ手段に書き込まれるデータの書き込み位置を、そのデータが前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段を有する。

【0020】前記書き込み制御手段は前記キューレジスタ手段に書き込まれたデータの最上位の位置、及び、最後尾の位置を記憶する回路を有することが望ましい。

【0021】本発明の他の実施態様によれば、前記主記憶部が複数のバンクからなり、前記複数のバンクのそれぞれに格納されたデータを、命令実行の順序に従って前記キューレジスタ手段に書き込む順序を制御する書き込み制御手段を有する。

【0022】前記書き込み制御手段は、前記各バンクに対応する複数のデータバッファと、前記複数のバンクから前記複数のデータバッファへのデータの書き込みを制御する書き込み制御回路と、前記キューレジスタ手段へ転送すべきデータが格納されている前記主記憶のバンクの番号を、命令実行の順序に従って記憶するバンク順序記憶回路と、前記複数のデータバッファからのデータを、前記バンク順序記憶回路に記憶されているバンクの番号に従って選択し、前記キューレジスタ手段に書き込むための選択回路とを有することが望ましい。

【0023】本発明の他の実施態様によれば、前記キューレジスタ手段は前記命令処理ユニットに接続される先入れ先出し型の第1のキューレジスタと、この第1のキューレジスタと前記主記憶部との間に接続される先入れ先出し型の第2のキューレジスタとから成る。

【0024】前記主記憶部から前記第2のキューレジスタに書き込まれるデータの書き込み位置を、そのデータ

が前記命令処理ユニットで処理される順序に従って制御する書き込み制御手段を有することが望ましい。

【0025】

【作用】本発明の情報処理装置、及び、情報処理システムでは、命令処理ユニットがプリロード命令を実行すると、指定されたデータが主記憶部からキューレジスタに転送される。データを処理する十分前にプリロード命令を実行し、実際にデータを処理する時にキューレジスタからデータを読み出して処理することにより、高速に処理を行うことができる。

【0026】また、命令処理ユニットが実行する命令セットがキューレジスタからレジスタへデータを転送するキュー転送命令を含む場合には、命令処理ユニットがプリロード命令を実行すると主記憶部から転送されてきたデータはキューレジスタの最後尾に書き込まれ、命令処理ユニットがキュー転送命令を実行するとキューレジスタの最初のデータがレジスタに転送されキューレジスタから削除される。データを処理する十分前にプリロード命令を実行し、実際にデータを処理する時にキュー転送命令を実行して必要なデータをレジスタに転送することにより、高速に処理を行うことができる。

【0027】また、プリロード命令の中に、主記憶からキューレジスタへのデータの転送とキューレジスタからレジスタへのデータの転送とを一つの命令で行う複合プリロード命令を含む場合には、命令処理ユニットが複合プリロード命令を実行すると、主記憶部から転送されてきたデータはキューレジスタの最後尾に書き込まれ、それと同時にキューレジスタの最初のデータがレジスタに転送されキューレジスタから削除される。現在処理しようとしているデータのキューレジスタからレジスタへの転送と、将来処理するデータの主記憶部からキューレジスタへの転送を一命令で実行することにより、プリロード命令とキュー転送命令の二つの命令を実行する場合に比べ処理に必要な命令数が減少するため、処理時間が短縮され、性能が向上する。

【0028】また、命令処理ユニットが実行する命令セットが、キューレジスタからデータを削除するキュー削除命令を含む場合には、命令処理ユニットがキュー削除命令を実行すると、キューレジスタに格納されているデータのうち指定された数のデータが削除される。将来必要となることを予想して予めプリロード命令でキューレジスタに転送されていたデータが必要ないことが判明した場合には、キュー削除命令で複数のデータを同時にキューレジスタから削除することにより、キュー転送命令を複数回実行してキューレジスタからデータを削除する場合に比べ処理に必要な命令数が減少するため、処理時間が短縮され、性能が向上する。

【0029】また、命令処理ユニットが実行する命令セットが、キューレジスタに格納されたデータを用いて演算器で演算を行うキュー演算命令を含む場合には、命令

処理ユニットがキュー演算命令を実行すると、キューレジスタの最初のデータがキューレジスタから削除され、演算器に転送され、演算器で処理される。予めプリロード命令でキューレジスタに転送されていたデータを処理する場合にキュー演算命令を実行することにより、キュー転送命令でレジスタに転送してから演算命令を実行する場合に比べ処理に必要な命令数が減少するため、処理時間が短縮され、性能が向上する。

【0030】また、キュー演算命令の中に、キューレジスタに格納された複数のデータの間の演算を行う複数データキュー演算命令を含む場合には、命令処理ユニットが複数データキュー演算命令を実行すると、キューレジスタに格納された複数のデータがキューレジスタから削除され、演算器に転送され、演算器で処理される。予めプリロード命令でキューレジスタに転送されていた複数のデータの間の演算を実行する場合に、複数データキュー演算命令を実行することにより、一部のデータをキュー転送命令でレジスタに転送した後にキュー演算命令を実行する場合に比べ処理に必要な命令数が減少するため、処理時間が短縮され、性能が向上する。

【0031】また、キュー演算命令の中に、実行後に演算に使用したデータをキューレジスタから削除するキュー演算削除命令と、削除しないキュー演算非削除命令を含む場合には、命令処理ユニットがキュー演算非削除命令を実行すると、キューレジスタに格納されたデータがキューレジスタから削除されずに演算器に転送され、演算器で処理される。キューレジスタに格納されたデータに複数の演算を実行する場合に、最後の演算のみにキュー演算削除命令を実行しそれ以外の演算にはキュー演算非削除命令を実行することにより、キュー転送命令を実行してキューレジスタからレジスタにデータを転送した後に複数の演算命令を実行する場合に比べ処理に必要な命令数が減少するため、処理時間が短縮され、性能が向上する。

【0032】また、主記憶に接続された複数のキューレジスタと、各キューレジスタと演算器を接続する複数のデータバスとを有する場合には、命令処理ユニットは複数のキューレジスタに格納されたデータの間の演算を行う複数キュー演算命令を実行する。主記憶上の複数のデータの間の演算を行う場合に、各データを予めプリロード命令により主記憶から別々のキューレジスタへ転送しておき、複数キュー演算命令を実行することにより、一つのキューレジスタのみを使用しキュー転送命令とキュー演算命令を実行する場合に比べ処理に必要な命令数が減少するため、処理時間が短縮され、性能が向上する。

【0033】また、命令処理ユニットが複数の演算器と、演算器それぞれに接続された複数のレジスタを有し、各レジスタそれぞれに接続された複数のキューレジスタを有する場合には、命令処理ユニットは主記憶から各キューレジスタにデータを転送するプリロード命令を

実行する。各演算器で処理を行う時には、対応するキューレジスタから対応するレジスタにデータを転送して処理することにより、高速に処理を行うことができる。

【0034】また、命令処理ユニットが複数の演算器と、演算器それぞれに接続された複数のレジスタを有し、単一のキューレジスタと複数のレジスタを接続するデータバスを有する場合には、命令処理ユニットは主記憶からキューレジスタにデータを転送するプリロード命令を実行する。各演算器で処理を行う時には、単一のキューレジスタから演算器に対応したレジスタにデータを転送して処理することにより、高速に処理を行うことができる。

【0035】また、キューレジスタが複数のデータを格納する領域を有し、主記憶部から前記キューレジスタに転送されたデータをプリロード命令の順番に並べ変える順序保証回路を有する場合には、主記憶部から転送されてくるデータの順番がプリロード命令の順番と異なっている場合でも、順序保証回路がプリロード命令の順番に並べ変えてキューレジスタに書き込む。これによって、プリロード命令の順番と異なった順番で主記憶部からデータを読み出した方が高速に処理できる場合に、最適な順番で主記憶部からデータを読み出すことが可能となり、プリロード命令の順番と同じ順番で主記憶部からデータを読み出す場合に比べ高速に処理を行うことができる。

【0036】また、キューレジスタへのデータの書き込みを制御する書き込み制御回路と、次に実行するプリロード命令のデータを格納すべきキューレジスタ内の場所を示すボトムポイントと、ボトムポイントから主記憶部に制御情報を転送するためのデータバスと、主記憶部から書き込み制御回路に制御情報を転送するためのデータバスとを有する場合には、各プリロード命令を実行する時にそのプリロード命令のデータを書き込むべき場所を示す制御情報がボトムポイントから主記憶部に転送される。主記憶部からキューレジスタにデータが転送される時にはそのデータを書き込むべきキューレジスタの場所を示す制御情報が主記憶部から書き込み制御回路に転送され、書き込み制御回路は主記憶部から転送されてきた制御情報に従ってキューレジスタにデータを書き込む。これによって、主記憶部から転送されてくるデータの順番がプリロード命令の順番と異なっている場合でも、キューレジスタにはプリロード命令の順番にデータが書き込まれる。

【0037】また、主記憶が複数のバンクより構成され、主記憶部の各バンクに対応する複数のデータバッファと、複数のデータバッファからのデータを選択する選択回路と、データバッファへのデータの書き込みを制御する書き込み制御回路と、プリロード命令で転送すべきデータが格納されている主記憶部のバンクの番号を記憶するバンク順序記憶回路と、主記憶部から書き込み制御回路に制御情報を転送するデータバスと、バンク順序記憶

回路から選択回路に制御情報を転送するためのデータバスとを有する場合には、各プリロード命令を実行する時にそのプリロード命令のデータが格納されている主記憶のバンクの番号がプリロード命令の実行の順番にバンク順序記憶回路に格納される。主記憶部からデータが転送されてくるときには、そのデータが格納されていた主記憶のバンクの番号が書き込み制御回路に転送される。書き込み制御回路は、主記憶部から受け取ったバンクの番号に対応するデータバッファにデータを書き込む。選択回路は、バンク順序記憶回路に格納されたバンク番号の

順番に従って、複数のデータバッファからのデータを選択し、キューレジスタに転送する。これによって、主記憶から転送されてくるデータの順番がプリロード命令の順番と異なっている場合でも、キューレジスタにはプリロード命令の順番にデータが書き込まれる。

【0038】また、キューレジスタが二つの部分よりなり、キューレジスタの第一の部分は命令処理ユニットに接続され、キューレジスタの第二の部分は主記憶部に接続され、順序保証回路がキューレジスタの第二の部分に接続されている場合には、順序保証回路は、主記憶から転送されてきたデータをプリロード命令の実行順番に並べ変えてキューレジスタの第二の部分に書き込む。キューレジスタの第二の部分に書き込まれたデータはプリロード命令の実行の順番にキューレジスタの第一の部分に転送される。これによって、命令処理ユニットに接続されたキューレジスタの第一の部分は構成が簡単になり高速な動作が可能である。一方、キューレジスタの第二の部分はキューレジスタの第一の部分に比べて低速であるが、一度に複数のデータをキューレジスタの第二の部分から第一の部分に転送することにより、命令処理ユニットが必要とする速度でデータを転送することが可能となる。従って、キューレジスタが一つの部分よりなりかつ順序保証回路を有する場合に比べ、高速に処理を行うことが可能となる。

【0039】

【実施例】以下、本発明の実施例を図面を用いて説明する。

【0040】図16は、本発明の第一の実施例で実行する命令を示した図である。図16に示した命令のうち、項番7のプリロード命令、項番8のキュー転送命令、項番9の複合プリロード命令、項番10のキュー削除命令、項番11のキューデータ数読み出し命令は、本発明の特徴であるキューレジスタに関する命令である。項番9の複合プリロード命令は、項番7のプリロード命令の処理と項番8のキュー転送命令の処理とを1命令で行う複合命令である。

【0041】図17は、図16に示した命令のうち、キューレジスタに関する命令を実行した時のキューレジスタの状態の変化を示した図である。図17(a)は命令列の例であり、命令1から命令8の順に実行される。

図17(b)は、図17(a)の命令列を実行した時のキューレジスタの状態の変化を示した図である。キューレジスタは5つの格納領域を有し、5つのデータを格納することが可能である。このうち、図17(b)上で一番上に記述された格納領域を最上部と呼ぶ。また、データの入っていない格納領域のうち一番上の格納領域を最後尾と呼ぶ。図17(b)では、最後尾を記号Bで示してある。初期状態では全ての格納領域は空であり、最後尾と最上部が等しい。命令1のプリロード命令を実行すると、データ1が最後尾に書き込まれ、最後尾の位置が一つ下に移動する。命令2のプリロード命令を実行すると、データ2が最後尾に書き込まれ、最後尾の位置が一つ下に移動する。命令3のプリロード命令を実行すると、データ3が最後尾に書き込まれ、最後尾の位置が一つ下に移動する。命令4のプリロード命令を実行すると、データ4が最後尾に書き込まれ、最後尾の位置が一つ下に移動する。命令5のキュー転送命令を実行すると、最上部のデータ(データ1)がレジスタtに転送され、キューレジスタから削除される。その際、最上部以外の格納領域のデータが一つ上の格納領域に移動し、それに伴い最後尾の位置も一つ上に移動する。命令6のキューデータ数読み出し命令を実行すると、キューレジスタに格納されているデータの数(ここでは3)がレジスタtに書き込まれる。命令7のキュー削除命令を実行すると、キューレジスタの最上部から順に指定された数(ここでは2個)のデータが削除される。その際、残りのデータは削除されたデータの分だけ上に移動し、それに伴い最後尾の位置も上に移動する。命令8の複合プリロード命令を実行すると、データ5が最後尾に書き込まれ、同時に最上部のデータ(データ4)がレジスタtに転送され、キューレジスタから削除される。その際、最上部以外の格納領域のデータが一つ上の格納領域に移動する。最後尾の位置は変化しない。以上の説明では、簡単のため、1つの命令の処理が全て終了してから次の命令の処理を開始するものとして説明した。実際の情報処理装置では、通常パイプライン的に処理が行われる。従って、例えば、命令4のデータがキューレジスタに書き込まれる前に命令5の処理が行われる可能性もある。

【0042】図1は、本発明の第一の実施例のブロック図である。102は命令処理ユニット、142は命令デコーダ、104は演算器、106はレジスタ、144はキャッシュ制御ユニット、146は命令フェッチユニット、108はキャッシュメモリ、110はキューレジスタ、148は主記憶制御ユニット、112は主記憶、122、124、126、128、130および132はデータバスである。キューレジスタ110は、10個の格納領域を有する。キャッシュメモリ108は、データ及び命令を格納している。実施例では、データ及び命令を格納している場合を説明するが、データ及び命令とそれぞれを個別に格納するデータキャッシュ及び命令キ

キャッシュを用いても良い。

【0043】命令フェッチユニット146は、キャッシュ制御ユニット144に命令コードの読み出しを要求する。命令フェッチユニット146の要求した命令コードがキャッシュメモリ108に存在する時には、キャッシュ制御ユニット144はデータバス128を通してキャッシュメモリ108から命令デコード142に命令コードを転送する。命令フェッチユニット146の要求した命令コードがキャッシュメモリ108に無い時には、キャッシュ制御ユニット144は、主記憶制御ユニット148に命令コードの読み出しを要求する。主記憶制御ユニット148は、主記憶112から命令コードを読み出し、データバス132を通してキャッシュメモリ108に転送する。キャッシュ制御ユニット144は、主記憶制御ユニット148から転送されてきた命令コードをキャッシュメモリ108に書き込み、同時にデータバス128を通して命令デコード142に転送する。

【0044】命令デコード142は、キャッシュメモリ108から転送されてきた命令コードを解読し、命令コードの種類に応じた処理を行う。

【0045】命令コードが演算命令であった場合には、命令デコード142は、データバス122、124を通してレジスタ106から演算器104にデータを転送し、演算器104で演算を行う。演算が終了すると、データバス126を通して演算器104からレジスタ106に演算結果を転送し、レジスタ106に書き込む。

【0046】命令コードがロード命令であった場合には、命令デコード142は、データバス122、124を通してレジスタ106から演算器104にデータを転送し、演算器104でアドレス計算を行う。アドレス計算が終了すると、データバス126を通してアドレスをキャッシュ制御ユニット144に転送し、キャッシュ制御ユニット144にデータの読み出しを要求する。命令デコード142の要求したデータがキャッシュメモリ108に存在する時には、キャッシュ制御ユニット144はデータバス128を通してキャッシュメモリ108からレジスタ106にデータを転送する。命令デコード142の要求したデータがキャッシュメモリ108に無い時には、キャッシュ制御ユニット144は、主記憶制御ユニット148にデータの読み出しを要求する。主記憶制御ユニット148は、主記憶112からデータを読み出し、データバス132を通してキャッシュメモリ108に転送する。キャッシュ制御ユニット144は、主記憶制御ユニット148から転送されてきたデータをキャッシュメモリ108に書き込み、同時にデータバス128を通してレジスタ106に転送する。

【0047】命令コードがストア命令であった場合には、命令デコード142は、データバス122、124を通してレジスタ106から演算器104にデータを転送し、演算器104でアドレス計算を行う。アドレス計

算が終了すると、データバス126を通してアドレスをキャッシュ制御ユニット144に転送し、同時にデータバス128を通してデータをキャッシュメモリ108に転送して、キャッシュ制御ユニット144にデータの書き込みを要求する。命令デコード142の要求したアドレスのデータがキャッシュメモリ108に存在する時には、キャッシュ制御ユニット144はデータバス128を通してレジスタ106から転送されてきたデータをキャッシュメモリ108に書き込む。命令デコード142の要求したアドレスのデータがキャッシュメモリ108に無い時には、キャッシュ制御ユニット144は、主記憶制御ユニット148にデータの読み出しを要求する。主記憶制御ユニット148は、主記憶112からデータを読み出し、データバス132を通してキャッシュメモリ108に転送する。キャッシュ制御ユニット144は、主記憶制御ユニット148から転送されてきたデータをキャッシュメモリ108に書き込み、書き込み終了後に、データバス128を通してレジスタ106から転送されてきたデータをキャッシュメモリ108に書き込む。

【0048】命令コードが代入命令であった場合には、命令デコード142は、命令コードに指定された定数値をレジスタ106に書き込む。

【0049】命令コードが条件分岐命令であった場合には、命令デコード142は、データバス122、124を通してレジスタ106から演算器104にデータを転送し、演算器104で比較を行う。比較結果が指定された条件を満たした場合には、命令デコード142は、命令フェッチユニット146に分岐先の命令コードの読み出しを指示する。

【0050】命令コードがプリロード命令であった場合には、命令デコード142は、キューレジスタ110に空の格納領域があるかどうかを調べる。キューレジスタ110に空の格納領域がある場合は、命令デコード142は、データバス122、124を通してレジスタ106から演算器104にデータを転送し、演算器104でアドレス計算を行う。アドレス計算が終了すると、データバス126を通してアドレスをキャッシュ制御ユニット144に転送し、キャッシュ制御ユニット144にデータの読み出しを要求する。キャッシュ制御ユニット144は、主記憶制御ユニット148にデータの読み出しを要求する。主記憶制御ユニット148は、主記憶112からデータを読み出し、データバス132を通してキューレジスタ110に転送し、キューレジスタ110の最後尾に書き込む。キューレジスタ110に空の格納領域が無い場合は、例外割り込みが発生し、命令デコード142は、命令フェッチユニット146に、例外処理ルーチンのアドレスに対応する命令コードの読み出しを指示する。

【0051】命令コードがキュー転送命令であった場合

には、命令デコーダ142は、キューレジスタ110の最上部にデータが存在するかどうかを調べる。キューレジスタ110の最上部にデータが存在する場合は、そのデータをデータバス130を通してレジスタ106に転送し、レジスタ106に書き込む。キューレジスタ110の最上部にデータが存在しない場合は、命令デコーダ142は、主記憶制御ユニット148が先行するプリロード命令の読み出し処理中であるかどうかを調べる。主記憶制御ユニット148が先行するプリロード命令の読み出し処理中である場合には、命令デコーダ142は、主記憶制御ユニット148がキューレジスタ110の最上部にデータを書き込むのを待って、そのデータをデータバス130を通してレジスタ106に転送し、レジスタ106に書き込む。主記憶制御ユニット148が先行するプリロード命令の読み出し処理中でない場合には、例外割り込みが発生し、命令デコーダ142は、命令フェッチユニット146に、例外処理ルーチンのアドレスに対応する命令コードの読み出しを指示する。

【0052】命令コードが複合プリロード命令であった場合には、命令デコーダ142は、プリロード命令の時の処理と、キュー転送命令の時の処理の両方を行う。

【0053】命令コードがキュー削除命令であった場合には、命令デコーダ142は、命令コードで指定された数のデータがキューレジスタ110に存在するかどうかを調べる。キューレジスタ110に指定された数のデータが存在する場合は、それらのデータをキューレジスタ110から削除する。キューレジスタ110に指定された数のデータが存在しない場合は、命令デコーダ142は、主記憶制御ユニット148が先行するプリロード命令の読み出し処理中であるかどうかを調べる。主記憶制御ユニット148が先行するプリロード命令の読み出し処理中である場合には、命令デコーダ142は、主記憶制御ユニット148が先行するプリロード命令の読み出し処理を終了するのを待って、指定された数のデータをキューレジスタ110から削除する。主記憶制御ユニット148が先行するプリロード命令の読み出し処理を全て終了しても指定された数のデータがキューレジスタ110に存在しない場合には、例外割り込みが発生し、命令デコーダ142は、命令フェッチユニット146に、例外処理ルーチンのアドレスに対応する命令コードの読み出しを指示する。

【0054】命令コードがキューデータ数読み出し命令であった場合には、命令デコーダ142は、キューレジスタ110から読み出し可能なデータの数を調べ、レジスタ106に書き込む。ここで、キューレジスタ110から読み出し可能なデータの数は、キューレジスタ110に存在するデータの数と、主記憶制御ユニット148が読み出し処理中の先行プリロード命令の数の和に等しい。

【0055】図2は、図16に示した命令のうち、キュー

ーレジスタに関する命令の命令コードを示した図である。各命令コードは、32ビットの2進数で表わされる。図2では、説明の都合上、命令コードの各ビットに最上位ビットから順に0から31の番号を付けてある。

【0056】図2(a)は、プリロード命令、キュー転送命令および複合プリロード命令の命令コードを一つの図にまとめたものである。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP1の時、その命令コードはプリロード命令、キュー転送命令または複合プリロード命令として処理される。pフィールド(ビット10)は、プリロードの処理を行うかどうかを示すフィールドである。mフィールド(ビット11)は、キュー転送の処理を行うかどうかを示すフィールドである。pフィールドが1、mフィールドが0の時、この命令はプリロード命令として処理される。pフィールドが0、mフィールドが1の時、この命令はキュー転送命令として処理される。pフィールドが1、mフィールドが1の時、この命令は複合プリロード命令として処理される。bフィールド(ビット17からビット21)及びxフィールド(ビット22からビット26)は、プリロードの処理を行う時のアドレス計算に使用するレジスタの番号を指定するフィールドである。bフィールドで指定されたレジスタの値と、xフィールドで指定されたレジスタの値を加算した値が、アドレスとなる。tフィールド(ビット27からビット31)は、キュー転送の処理を行う時にデータを書き込むレジスタの番号を指定するフィールドである。

【0057】図2(b)は、キュー削除命令およびキューデータ数読み出し命令の命令コードである。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP2の時、その命令コードはキュー削除命令またはキューデータ数読み出し命令として処理される。kフィールド(ビット10からビット11)は、処理の種類を示すフィールドである。kフィールドが0の時、この命令はキュー削除命令として処理される。kフィールドが1の時、この命令はキューデータ数読み出し命令として処理される。nフィールド(ビット12からビット18)は、キュー削除命令を実行する時にキューレジスタから削除するデータの数を指定するためのフィールドである。tフィールド(ビット27からビット31)は、キューデータ数読み出し命令を実行する時に、キューレジスタから読み出し可能なデータの個数を書き込むレジスタの番号を指定するフィールドである。

【0058】図18は、キューデータ数読み出し命令の使用法の例を説明する図である。図は、複数のユーザプログラム、プログラム1とプログラム2を時分割で処理する様子を示したものであり、キューデータ数読み出し命令はキューレジスタ退避処理の際に使用される。

【0059】まず最初に、処理の流れの概略を説明す

る。時刻 t_1 にタイマ割り込みによってプログラム 1 の処理が中断され、OS の処理が開始される。時刻 t_1 から t_2 の間に OS がプログラム 1 からプログラム 2 への切り換え処理を行う。その際、キューレジスタに残されたデータをメモリ空間に設けられた退避領域に保存しておく必要がある。これがキューレジスタ退避処理である。時刻 t_2 から時刻 t_3 の間、プログラム 2 の処理が行われる。時刻 t_3 にタイマ割り込みによってプログラム 2 の処理が中断され、再び OS の処理が開始される。時刻 t_3 から t_4 の間に OS がプログラム 2 からプログラム 1 への切り換え処理を行う。その際、メモリ空間の退避領域に保存されていたデータをキューレジスタに戻す必要がある。これがキューレジスタ回復処理である。時刻 t_4 からプログラム 1 の処理が再開される。

【0060】次に、キューレジスタ退避処理について説明する。1 行目のキューデータ数読み出し命令を実行すると、キューレジスタに入っているデータの数がレジスタ r_3 に書き込まれる。2 行目の代入命令を実行すると、レジスタ r_1 に退避領域の先頭のアドレスが書き込まれる。3 行目の代入命令を実行すると、レジスタ r_2 に 0 が書き込まれる。4 行目のストア命令を実行すると、退避領域の先頭に、キューレジスタに入っているデータの数が書き込まれる。5 行目の条件分岐命令を実行すると、キューレジスタに入っているデータの数が 0 の時のみ 11 行目に分岐し、処理を終了する。キューレジスタに入っているデータの数が 1 以上の時は、6 行目の処理に進む。6 行目の演算命令を実行すると、レジスタ r_2 に 1 が加算され、レジスタ r_2 に書き込まれる。7 行目から 10 行目はループになっており、キューレジスタに入っているデータの数だけ繰り返される。7 行目のキュー転送命令と 8 行目のストア命令を実行すると、キューレジスタの最上部のデータがレジスタ r_0 を経由して退避領域に書き込まれ、キューレジスタから削除される。9 行目の演算命令を実行すると、レジスタ r_2 に 1 が加算され、レジスタ r_2 に書き込まれる。10 行目の条件分岐命令は、キューレジスタにデータが残っているかどうかを調べる条件分岐命令である。キューレジスタにデータが残っている場合は、7 行目に分岐し、処理を続ける。キューレジスタにデータが残っていない場合は、11 行目に進み、処理を終了する。

【0061】次に、キューレジスタ回復処理について説明する。1 行目の代入命令を実行すると、レジスタ r_1 に退避領域の先頭のアドレスが書き込まれる。2 行目の代入命令を実行すると、レジスタ r_2 に 0 が書き込まれる。3 行目のロード命令を実行すると、退避領域の先頭のデータが、レジスタ r_3 に書き込まれる。退避領域の先頭には、キューレジスタ退避処理を行った時にキューレジスタから退避領域に転送したデータの数が格納されているため、レジスタ r_3 に書き込まれた値は、退避領域からキューレジスタに転送すべきデータの数を示すこ

とになる。4 行目の条件分岐命令を実行すると、退避領域からキューレジスタに転送すべきデータの数が 0 の時のみ 9 行目に分岐し、処理を終了する。退避領域からキューレジスタへ転送すべきデータの数が 1 以上の時は、5 行目に進む。5 行目の演算命令を実行すると、レジスタ r_2 に 1 が加算され、レジスタ r_2 に書き込まれる。6 行目から 8 行目はループになっており、退避領域からキューレジスタに転送すべきデータの数だけ繰り返される。6 行目のプリロード命令を実行すると、退避領域のデータがキューレジスタの最後尾に転送される。7 行目の演算命令を実行すると、レジスタ r_2 に 1 が加算され、レジスタ r_2 に書き込まれる。8 行目の条件分岐命令は、退避領域からキューレジスタに必要な回数だけデータが転送されたかどうかを調べる条件分岐命令である。転送すべきデータが残っている場合は、6 行目に分岐し、処理を続ける。退避領域からキューレジスタに必要な回数だけデータが転送された場合は、9 行目に進み、処理を終了する。以上の処理によって、時刻 t_4 のキューレジスタの状態は、時刻 t_1 のキューレジスタの状態と同じになるため、 t_4 以降のプログラム 1 の処理が正しく行われることが保証される。

【0062】図 3 は、キューレジスタに関係する命令を使用したプログラムの例を示した図である。処理内容は要素数 100 の配列 x と y の加算を行い、要素数 100 の配列 z に書き込む、というものである。以下、配列 x の各要素を $x(1) \sim x(100)$ 、配列 y の各要素を $y(1) \sim y(100)$ 、配列 z の各要素を $z(1) \sim z(100)$ と記述することにする。1 行目の代入命令を実行すると、レジスタ r_5 に $x(1)$ のアドレスから 1 を減じた数が書き込まれる。2 行目の代入命令を実行すると、レジスタ r_6 に $y(1)$ のアドレスから 1 を減じた数が書き込まれる。3 行目の代入命令を実行すると、レジスタ r_0 に 1 が書き込まれる。以下の命令において、レジスタ r_0 は配列 x 、及び配列 y の添字として使用する。4 行目の代入命令を実行すると、レジスタ r_8 に 3 が書き込まれる。5 行目から 8 行目は、第一のループであり、3 回実行される。5 行目及び 6 行目の命令はプリロード命令であり、配列 x 、及び配列 y のデータをキューレジスタに転送するものである。これらの命令を 3 回実行すると、 $x(1)$ と $y(1)$ 、 $x(2)$ と $y(2)$ 、 $x(3)$ と $y(3)$ のデータがキューレジスタに書き込まれる。7 行目の演算命令を実行すると、レジスタ r_0 に 1 が加算されレジスタ r_0 に書き込まれる。8 行目の条件分岐命令を実行すると、第一のループが 3 回実行されるまでは 5 行目に分岐し、3 回実行されると 9 行目に進む。9 行目の代入命令を実行すると、レジスタ r_7 に $z(1)$ のアドレスから 1 を減じた数が書き込まれる。10 行目の代入命令を実行すると、レジスタ r_1 に 1 が書き込まれる。以下の命令において、レジスタ r_1 は配列 z の添字として使用する。11 行目の代入命令を実行すると、レ

レジスタ r_8 に100が書き込まれる。12行目から18行目は、第二のループであり、97回実行される。以下に、このループを1 ($i=1 \sim 97$) 回目に実行する時の処理を説明する。12行目の複合プリロード命令を実行すると、 $x(i+3)$ のデータがキューレジスタに書き込まれ、同時に、 $x(i)$ のデータがキューレジスタからレジスタ r_2 に転送される。13行目の複合プリロード命令を実行すると、 $y(i+3)$ のデータがキューレジスタに書き込まれ、同時に、 $y(i)$ のデータがキューレジスタからレジスタ r_3 に転送される。14行目の演算命令を実行すると、 $x(i)$ のデータと $y(i)$ のデータが加算され、レジスタ r_4 に書き込まれる。15行目のストア命令を実行すると、14行目の演算命令の加算結果が、 $z(i)$ に書き込まれる。16行目の演算命令を実行すると、レジスタ r_0 に1が加算されレジスタ r_0 に書き込まれる。17行目の演算命令を実行すると、レジスタ r_1 に1が加算されレジスタ r_1 に書き込まれる。18行目の条件分岐命令を実行すると、第二のループが97回実行されるまでは12行目に分岐する。第二のループが97回実行されると、19行目に進む。19行目から24行目は第三のループであり、3回実行される。以下、このループが1回目に実行されるとき処理を説明する。2回目、3回目の処理も同様である。19行目のキュー転送命令を実行すると、 $x(98)$ のデータがキューレジスタからレジスタ r_2 に転送される。同様に、20行目のキュー転送命令を実行すると、 $y(98)$ のデータがキューレジスタからレジスタ r_3 に転送される。21行目の演算命令を実行すると、 $x(98)$ のデータと $y(98)$ のデータが加算され、レジスタ r_4 に書き込まれる。22行目のストア命令を実行すると、21行目の命令の加算結果が、 $z(98)$ に書き込まれる。23行目の演算命令を実行すると、レジスタ r_1 に1が加算されレジスタ r_1 に書き込まれる。24行目の条件分岐命令を実行すると、第三のループが3回実行されるまでは19行目に分岐する。第三のループが3回実行されると、処理を終了する。

【0063】図4は、図3の命令列を図1の情報処理装置で実行したときの処理の流れを示した図である。サイクル1からサイクル4では、図3の命令列の1行目から4行目の代入命令の処理が行われる。サイクル5から40
サイクル8は、図3の命令列の5行目から8行目のループの一回目の処理に相当する。サイクル5では、図3の命令列の5行目のプリロード命令の処理が行われ、主記憶制御ユニットに対して $x(1)$ のデータの読み出し要求が行われる。主記憶の読み出しには10サイクル要するため、 $x(1)$ のデータはサイクル15にキューレジスタに書き込まれる。サイクル6では、図3の命令列の6行目のプリロード命令の処理が行われ、主記憶制御ユニットに $y(1)$ のデータの読み出し要求が出される。 $y(1)$ のデータはサイクル16にキューレジスタに書き

込まれる。サイクル7では演算命令、サイクル8では条件分岐命令の処理が行われる。サイクル9からサイクル12は、図3の命令列の5行目から8行目のループの二回目の処理に相当する。サイクル9及びサイクル10では、主記憶制御ユニットに対して $x(2)$ 及び $y(2)$ のデータの読み出し要求が行われ、対応するデータがサイクル19及びサイクル20にキューレジスタに書き込まれる。サイクル11では演算命令、サイクル12では条件分岐命令の処理が行われる。サイクル13からサイクル16についても同様であり、 $x(3)$ 及び $y(3)$ のデータがサイクル23及びサイクル24にキューレジスタに書き込まれる。サイクル17からサイクル19では、代入命令の処理が行われる。サイクル20からサイクル26は、図3の命令列の12行目から18行目のループの一回目の処理に相当する。サイクル20では、図3の命令列の12行目の複合プリロード命令の処理が行われ、主記憶制御ユニットに $x(4)$ のデータの読み出し要求が出される。それと同時に、 $x(1)$ のデータがキューレジスタからレジスタ r_2 に転送される。 $x(1)$ のデータは、サイクル15にキューレジスタに書き込まれているため、主記憶の読み出しを待つことなく、次の命令の処理に移ることが可能である。 $x(4)$ のデータはサイクル30にキューレジスタに書き込まれる。サイクル21では、図3の命令列の13行目の複合プリロード命令の処理が行われ、主記憶制御ユニットに $y(4)$ のデータの読み出し要求が出される。それと同時に、 $y(1)$ のデータがキューレジスタからレジスタ r_3 に転送される。 $y(1)$ のデータは、サイクル16にキューレジスタに書き込まれているため、主記憶の読み出しを待つことなく、次の命令の処理に移ることが可能である。 $y(4)$ のデータはサイクル31にキューレジスタに書き込まれる。サイクル22では、サイクル20及びサイクル21にキューレジスタからレジスタ r_2 及び r_3 に転送されたデータの加算が行われる。サイクル23では、サイクル22の加算結果が主記憶に書き込まれる。サイクル24およびサイクル25では加算命令の処理が、サイクル26では条件分岐命令の処理が行われる。サイクル27から33、サイクル34から40は、図3の命令列の12行目から18行目のループの二回目、及び三回目の実行であり、サイクル20からサイクル26と同様の処理が行われる。以上説明した様に、キューレジスタを使用することにより、主記憶の読み出しにかかる時間(10サイクル)に影響されず、1サイクルに1命令の割合で命令の処理を行うことができる。

【0064】図5(a)は、本発明の第二の実施例を示した図である。502、504は選択回路、506は演算器、508はレジスタ、510はキューレジスタ、542は命令デコーダ、544はキャッシュ制御ユニット、512はキャッシュメモリ、546は命令フェッチユニット、548は主記憶制御ユニット、514は主記

憶、522, 524, 526, 528, 530および532はデータバスである。この実施例の特徴は、キューレジスタ510と演算器506を接続するデータバス522、およびキューレジスタ510からのデータとレジスタ508からのデータのどちらかを選択する選択回路502, 504を設けたことにある。演算器506はレジスタ508から読み出したデータの間だけでなく、レジスタ508から読み出したデータとキューレジスタ510から読み出したデータの間で演算を行うことが可能である。図5(b)は、図5(a)の情報処理装置で実行するキュー演算命令の命令コードの例である。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP3の時その命令はキュー演算命令として処理される。q1フィールド(ビット10)及びq2フィールド(ビット11)は、演算器の第一の入力、及び、第二の入力にレジスタから読み出したデータを用いるかキューレジスタから読み出したデータを用いるかを指定する。q1フィールドが0の時には、s1フィールド(ビット17からビット21)で指定された番号のレジスタの値が演算器の第一の入力として用いられる。q1フィールドが1の時には、キューレジスタの最上部のデータが演算器の第一の入力として用いられる。同様に、q2フィールドが0の時には、s2フィールド(ビット22からビット26)で指定された番号のレジスタの値が演算器の第二の入力として用いられ、q2フィールドが1の時には、キューレジスタの最上部のデータが演算器の第二の入力として用いられる。dフィールド(ビット12)は、演算に使用したキューレジスタのデータを削除するかどうかを指定するフィールドである。dフィールドが1の時には、演算に使用されたキューレジスタのデータはキューレジスタから削除される。dフィールドが0の時には、演算に使用されたキューレジスタのデータはキューレジスタから削除されず、次の演算で使用することが可能である。kフィールド(ビット13からビット16)は、演算器で行う演算の種類を示すフィールドである。tフィールド(ビット27からビット31)は、演算結果を書き込むレジスタの番号を示すフィールドである。図5(b)のキュー演算命令を用いれば、図3の命令列の20行目のキュー転送命令と21行目の演算命令の二命令を一命令に置き換えることができるため、処理すべき命令の数が減少し、性能が向上する。また、キューレジスタのデータを複数の演算命令で使用する必要がある場合には、dフィールドが0のキュー演算命令を使用すれば、キューレジスタのデータは削除されないため、次の演算で使用することが可能である。

【0065】図6(a)は、本発明の第三の実施例を示した図である。602, 604は選択回路、642は命令デコーダ、606は演算器、608はレジスタ、610はキューレジスタ、644はキャッシュ制御ユニッ

ト、646は命令フェッチユニット、612はキャッシュメモリ、648は主記憶制御ユニット、614は主記憶、622, 624, 626, 628, 630, 632および634はデータバスである。この実施例の特徴は、キューレジスタ610と演算器606を接続する2本のデータバス626, 628を設けたことにある。データバス628はキューレジスタ610の最上部のデータを演算器606に転送するためのデータバスであり、データバス626はキューレジスタ610の二番目のデータを演算器606に転送するためのデータバスである。図6(b)は、図6(a)の情報処理装置で実行する第一のキュー演算命令の命令コードの例である。このキュー演算命令は、キューレジスタの最初のデータのみを使用し、キューレジスタの二番目のデータは使用しない。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP4の時その命令は第一のキュー演算命令として処理される。それ以外のビットの意味は、図5(b)のキュー演算命令と同じである。図6(c)は、図6(a)の情報処理装置で実行する第二のキュー演算命令の命令コードの例である。このキュー演算命令は、キューレジスタの最上部のデータと二番目のデータの両方を使用する。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP5の時その命令は第二のキュー演算命令として処理される。ssフィールド(ビット10)は、キューレジスタの最上部のデータを演算器の第一の入力と第二の入力のどちらに用いるかを指定する。ssフィールドが0の時には、キューレジスタの最上部のデータが演算器の第一の入力として用いられ、キューレジスタの二番目のデータが演算器の第二の入力として用いられる。ssフィールドが1の時には、キューレジスタの二番目のデータが演算器の第一の入力として用いられ、キューレジスタの最上部のデータが演算器の第二の入力として用いられる。d1フィールド(ビット11)及びd2フィールド(ビット12)は、演算に使用したキューレジスタのデータを削除するかどうかを指定するフィールドである。d1フィールドが1の時にはキューレジスタの最上部のデータはキューレジスタから削除され、d1フィールドが0の時にはキューレジスタの最上部のデータはキューレジスタから削除されない。同様に、d2フィールドが1の時にはキューレジスタの二番目のデータはキューレジスタから削除され、d2フィールドが0の時にはキューレジスタの二番目のデータはキューレジスタから削除されない。kフィールド(ビット13からビット16)は、演算の種類を示すフィールドである。tフィールド(ビット27からビット31)は、演算結果を書き込むレジスタの番号を示すフィールドである。図6(c)のキュー演算命令を用いれば、図3の命令列の19行目、及び20行目のキュー転送命令と21行目の演算命令の三命令を一命令に置き換えるこ

とができるため、処理すべき命令の数が減少し、性能が向上する。また、キューレジスタのデータを複数の演算命令で使用する必要がある場合には、d1フィールドまたはd2フィールドが0のキュー演算命令を使用すればキューレジスタのデータは削除されないため、次の演算命令で使うことが可能である。

【0066】図7は、本発明の第四の実施例を示した図である。702、704は選択回路、742は命令デコーダ、706は演算器、708はレジスタ、711は第一のキューレジスタ、712は第二のキューレジスタ、744はキャッシュ制御ユニット、746は命令フェッチユニット、714はキャッシュメモリ、748は主記憶制御ユニット、716は主記憶、722、724、726、728、730、732および734はデータバスである。この実施例の特徴は、二つのキューレジスタ711、712を設けたことにある。

【0067】図8(a)は、図7の情報処理装置で実行するプリロード命令の命令コードの例である。図8

(a)のプリロード命令は、主記憶から転送してきたデータを二つのキューレジスタのどちらに書き込むかを指定するため、qビットを有する。qビットが0の時には第一のキューレジスタにデータが書き込まれ、qビットが1の時には第二のキューレジスタにデータが書き込まれる。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP6の時その命令はプリロード命令として処理される。この命令コードではキュー転送の処理は行わないため、ビット27からビット31は未使用ビットである。それ以外のビットの意味は、図2(a)のプリロード命令と同じである。図8(b)は、図7の情報処理装置で実行する第一のキュー演算命令の命令コードの例である。このキュー演算命令は、第一のキューレジスタのデータのみを使用し、第二のキューレジスタのデータは使用しない。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP7の時その命令は第一のキュー演算命令として処理される。それ以外のビットの意味は、図5(b)のキュー演算命令と同じである。図8

(b)の命令コードと同様の命令コードを用いれば、第二のキューレジスタのデータのみを使用し、第一のキューレジスタのデータは使用しないキュー演算命令をサポートすることも可能である。図8(c)は、図7の情報処理装置で実行する第二のキュー演算命令の命令コードの例である。このキュー演算命令は、第一のキューレジスタのデータと第二のキューレジスタのデータの両方を使用する。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP8の時その命令は第二のキュー演算命令として処理される。ssフィールド(ビット10)は、第一のキューレジスタから読み出したデータを演算器の第一の入力と第二の入力のどちらに用いるかを指定する。ssフィールドが0の

時には、第一のキューレジスタの最上部のデータが演算器の第一の入力として用いられ、第二のキューレジスタの最上部のデータが演算器の第二の入力として用いられる。ssフィールドが1の時には、第二のキューレジスタの最上部のデータが演算器の第一の入力として用いられ、第一のキューレジスタの最上部のデータが演算器の第二の入力として用いられる。d1フィールド(ビット11)及びd2フィールド(ビット12)は、演算に使用したキューレジスタのデータを削除するかどうかを指定するフィールドである。d1フィールドが1の時には第一のキューレジスタの最上部のデータは第一のキューレジスタから削除され、d1フィールドが0の時には第一のキューレジスタの最上部のデータは第一のキューレジスタから削除されない。同様に、d2フィールドが1の時には第二のキューレジスタの最上部のデータは第二のキューレジスタから削除され、d2フィールドが0の時には第二のキューレジスタの最上部のデータは第二のキューレジスタから削除されない。kフィールド(ビット13からビット16)は、演算の種類を示すフィールドである。tフィールド(ビット27からビット31)は、演算結果を書き込むレジスタの番号を示すフィールドである。図8(c)のキュー演算命令を用いれば、図6(c)のキュー演算命令を用いた場合と同様、図3の命令列の19行目、及び20行目のキュー転送命令と21行目の演算命令の三命令を一命令に置き換えることができるため、処理すべき命令の数が減少し、性能が向上する。また、第一のキューレジスタ及び第二のキューレジスタは、最上部のデータの読み出し、及び削除を行えばよく、二番目のデータの読み出し、及び削除を行う必要がないため、図7の情報処理装置では図6(a)の情報処理装置に比べてキューレジスタの構成が簡単になる。

【0068】図9(a)は、整数データ用と浮動小数点(FP)データ用に別々のレジスタ及び別々の演算器を有する情報処理装置に対して、本発明を適用した例である。952は命令デコーダ、902は整数演算器、904は整数レジスタ、906は浮動小数点演算器、908は浮動小数点レジスタ、954はキャッシュ制御ユニット、956は命令フェッチユニット、910はキャッシュメモリ、912は整数用キューレジスタ、914は浮動小数点用キューレジスタ、958は主記憶制御ユニット、916は主記憶、922、924、926、928、930、932、934、936、938、および940はデータバスである。本実施例は、整数用と浮動小数点用の二つのキューレジスタを有する特徴とする。整数用キューレジスタ912は整数レジスタ904に接続され、浮動小数点用キューレジスタ914は浮動小数点レジスタ908に接続されている。図9(b)は、図9(a)の情報処理装置で実行する複合プリロード命令の命令コードの例である。図9(b)の複合プリロード

命令は、整数データを扱うか浮動小数点データを扱うかを指定するため、pfビット（ビット12）及びmfビット（ビット13）を有する。pfビットが0の時には、主記憶から転送されてきたデータは整数用キューレジスタに書き込まれ、pfビットが1の時には、主記憶から転送されてきたデータは浮動小数点用キューレジスタに書き込まれる。mfビットが0の時には、整数用キューレジスタから整数レジスタにデータが転送され、mfビットが1の時には、浮動小数点用キューレジスタから浮動小数点レジスタにデータが転送される。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP9の時その命令は複合プリロード命令として処理される。それ以外のビットの意味は、図2（a）の複合プリロード命令と同じである。整数データ用と浮動小数点データ用の二つのキューレジスタを設けたことにより、整数データを処理する場合にも、浮動小数点データを処理する場合にも高速に処理を行うことが可能である。

【0069】図10（a）は、本発明の第六の実施例を示した図である。1042は命令デコーダ、1002は整数演算器、1004は整数レジスタ、1006は浮動小数点演算器、1008は浮動小数点レジスタ、1044はキャッシュ制御ユニット、1046は命令フェッチユニット、1010はキャッシュメモリ、1012はキューレジスタ、1048は主記憶制御ユニット、1014は主記憶、1022、1024、1026、1028、1030、1032、1034、1036および1038はデータバスである。本実施例の特徴は、整数データと浮動小数点データで一つのキューレジスタを共用する点にある。キューレジスタ1012は、整数レジスタ1004と浮動小数点レジスタ1008の両方に接続されている。図10（b）は、図10（a）の情報処理装置で実行する複合プリロード命令の命令コードの例である。図10（b）の複合プリロード命令は、整数レジスタにデータを転送するか浮動小数点レジスタにデータを転送するかを指定するため、fビット（ビット12）を有する。fビットが0の時には、キューレジスタから整数レジスタにデータが転送され、fビットが1の時には、キューレジスタから浮動小数点レジスタにデータが転送される。ビット0からビット9までは、命令の種類を示すフィールドであり、このフィールドがOP10の時その命令は複合プリロード命令として処理される。それ以外のビットの意味は、図2（a）の複合プリロード命令と同じである。整数データと浮動小数点データで一つのキューレジスタを共用することにより、図9（a）の情報処理装置に比べてハードウェア物量を削減することが可能である。

【0070】図11は、本発明の第七の実施例を示した図である。1152は命令デコーダ、1102は演算器、1104はレジスタ、1154はキャッシュ制御ユ

ニット、1156は命令フェッチユニット、1106はキャッシュメモリ、1108は順序保証回路、1110はキューレジスタ、1158は主記憶制御ユニット、1120は主記憶、1132、1134、1136、1138、1140、1142および1144はデータバスである。本実施例の特徴は、キューレジスタ1110内のデータがプリロード命令の実行順序と一致する様にデータの書き込みを制御する、順序保証回路1108を設けたことにある。主記憶1120は4つのバンクから構成されている。各バンクは容量を大きくするため比較的低速なメモリで構成されており、読み出しに10サイクル必要である。また、各バンクの読み出しはパイプライン的に処理され、同一バンクから連続してデータを読み出す場合は、4サイクルに一つのデータを読み出すことができる。更に、バンク0からバンク3の読み出しはパイプライン的に処理され、バンク0からバンク3を順次読み出す場合には1サイクルに1つのデータを読み出すことができる。キューレジスタ1110は、複数のデータを格納する領域を有する。データバス1144はキューレジスタ1110内にあるデータ格納領域全てに接続されている。順序保証回路1108は、主記憶1120から転送されてきたデータがプリロードの実行順序に並ぶように、キューレジスタ1110へのデータの書き込みを制御する。

【0071】図12は、図11の情報処理装置でプリロード命令の処理を行う時の処理の流れを示した図である。サイクル1からサイクル8にプリロード命令（a）から（h）の処理が行われ、主記憶の読み出し要求が出される。プリロード命令（a）のデータは、読み出し要求の10サイクル後にキューレジスタに書き込まれる。プリロード命令（b）は、プリロード命令（a）と同一バンクに対する読み出しであるため、プリロード命令（a）の4サイクル後でなければ読み出しができず、読み出し要求の14サイクル後にデータがキューレジスタに書き込まれる。一方、プリロード命令（c）は、プリロード命令（a）及び（b）とは別のバンクの読み出しであるため、プリロード命令（a）及び（b）の読み出しと並列に処理され、プリロード命令（b）より先にデータがキューレジスタに書き込まれる。この様に、プリロード命令（c）の読み出しをプリロード命令（b）の読み出しより先に行うことにより、高速に処理を行うことができる。同様に、プリロード命令（e）の読み出しはプリロード命令（b）及び（d）の読み出しより先に行われ、プリロード命令（g）の読み出しはプリロード命令（d）及び（f）の読み出しより先に行われる。この様に、主記憶の読み出しはプリロード命令の実行順序と異なるが、順序保証回路は、キューレジスタ内のデータがプリロード命令の実行順に並ぶ様にキューレジスタの書き込みを制御する。

【0072】図13は、本発明の第八の実施例を示した

図である。1352は命令デコーダ、1302は演算器、1304はレジスタ、1354はキャッシュ制御ユニット、1356は命令フェッチユニット、1306はキャッシュメモリ、1308、1310はポインタ、1312はキューレジスタの書き込み制御回路、1314はキューレジスタ、1358は主記憶制御ユニット、1320は主記憶、1332、1334、1336、1338、1340、1342、1344、1346および1348はデータバスである。トップポインタ1308はキューレジスタ1314の最上部の位置を記憶する回路である。キューレジスタ1314の最上部のデータを読み出すと、ポインタ1308に記憶された位置のデータが読み出される。キューレジスタ1314の最上部のデータを削除すると、ポインタ1308が次のデータの位置を示す値に更新される。ボトムポインタ1310はキューレジスタ1314の最後尾の位置を記憶する回路である。プリロード命令を実行すると、主記憶1320に対する読み出し要求に同期してポインタ1310に記憶された値がデータバス1342を通して主記憶制御ユニット1358に転送される。その後、ポインタ1310が次のデータの位置を示す値に更新される。主記憶制御ユニット1358は、主記憶1320から読み出したデータをデータバス1348に出力すると同時に、そのデータを書き込むべきキューレジスタ1314の位置をデータバス1344を通して書き込み制御回路1312に転送する。書き込み制御回路1312は、主記憶制御ユニット1358から受け取った情報に従って、キューレジスタ1314の書き込みを制御する。これによって、主記憶制御ユニット1358から転送されるデータの順番がプリロード命令の実行順番と異なっている、キューレジスタ1314内ではプリロード命令の実行順にデータを並べることが可能である。

【0073】図14は、本発明の第九の実施例を示した図である。1462は命令デコーダ、1402は演算器、1404はレジスタ、1464はキャッシュ制御ユニット、1466は命令フェッチユニット、1406はキャッシュメモリ、1410はバンク順序記憶回路、1408はキューレジスタ、1414は選択回路、1420～1423はデータバッファ、1412はデータバッファの書き込み制御回路、1468は主記憶制御ユニット、1430は主記憶、1442、1444、1446、1448、1450、1452、1454、1456および1458はデータバスである。主記憶1430は、図11の情報処理装置と同様、4つのバンクから構成されている。データバッファ1420～1423は、先入れ先出し型のデータバッファであり、それぞれ主記憶1430のバンク0～3に対応している。書き込み制御回路1412は、データバス1454を通して主記憶制御ユニット1468からデータバス1452に出力されたデータに対応する主記憶1430のバンクの番号を受け取り、

データバッファ1420～1423への書き込み制御信号をデータバス1458に出力する。バンク順序記憶回路1410は、各プリロード命令が読み出す主記憶1430のバンクの番号をプリロード命令の実行順序に対応付けて記憶する。選択回路1414は、データバス1456を通してバンク順序記憶回路1410から次のプリロード命令のバンクの番号を受け取り、対応するデータバッファ1420～1423のデータを選択する。キューレジスタ1408は先入れ先出し型のレジスタであり、選択回路1414から受け取ったデータを記憶する。バンク順序記憶回路1410に記憶されたバンクの番号に基づいて選択回路1414でデータを選択することにより、主記憶1430から読み出したデータの順番がプリロード命令の順番と異なっている、キューレジスタ1408にはプリロード命令の順番にデータが書き込まれる。

【0074】図15は、本発明の第十の実施例を示した図である。1514は命令処理ユニット、1552は命令デコーダ、1502は演算器、1504はレジスタ、1554はキャッシュ制御ユニット、1556は命令フェッチユニット、1506はキャッシュメモリ、1508、1510はレジスタ、1512は順序保証回路、1558は主記憶制御ユニット、1520は主記憶、1532、1534、1536、1538、1540、1542、1544および1546はデータバスである。本実施例の特徴は、キューレジスタが2つのレジスタ1508、1510に分割されていることである。レジスタ1508は少容量の先入れ先出し型のレジスタであり、高速に読み出すことができる。レジスタ1510は大容量のレジスタであり、順序保証回路1512によって制御されている。レジスタ1510はレジスタ1508に比べて読み出し速度が遅いが、一度に複数のデータを読み出すことにより、レジスタ1508に必要なデータを転送する。キューレジスタを2つのレジスタに分割したことにより、順序保証回路1512を設けたことによる処理速度の低下に影響されずに、命令処理ユニット1514へのデータ転送要求に高速に応答することが可能となる。また、データバス1546のデータ転送周波数を命令処理ユニット1514の動作周波数より低くすることが可能なため、レジスタ1510及び順序保証回路1512を別LSIとして構成することが容易となり、それによってLSIのチップサイズによる制約を気にせずレジスタ1510の大容量化を容易に図ることができる。

【0075】

【発明の効果】以上詳述した様に、本発明によれば、データが必要となるより十分前に主記憶からキューレジスタにデータを転送するプリロード命令を実行し、データを使用する時にキューレジスタからレジスタにデータを転送するキュー転送命令を実行することにより、低速な主記憶の読み出し時間に影響されずに高速に処理を行う

ことが可能となり、情報処理装置の性能を向上させることが可能である。

【図面の簡単な説明】

【図1】本発明の第一の実施例を示すブロック構成図である。

【図2】図1の情報処理装置で実行するプリロード命令、キュー転送命令、複合プリロード命令及びキュー削除命令、キューデータ数読み出し命令の命令コードの例を示す図である。

【図3】図2のプリロード命令、キュー転送命令、および複合プリロード命令を使用したプログラムの例を示す図である。

【図4】図3のプログラムを図1の情報処理装置で処理した時の処理の流れを示す図である。

【図5】本発明の第二の実施例のブロック構成図、及び、本実施例の情報処理装置で処理するキュー演算命令の命令コードの例を示す図である。

【図6】本発明の第三の実施例のブロック構成図、及び、本実施例の情報処理装置で処理するキュー演算命令の命令コードの例を示す図である。

【図7】本発明の第四の実施例のブロック構成図である。

【図8】図7の情報処理装置で実行するプリロード命令及びキュー演算命令の命令コードの例を示す図である。

【図9】本発明の第五の実施例のブロック構成図、及び、本実施例の情報処理装置で処理する複合プリロード命令の命令コードの例を示す図である。

【図10】本発明の第六の実施例のブロック構成図、及び、本実施例の情報処理装置で処理する複合プリロード命令の命令コードの例を示す図である。

【図11】本発明の第七の実施例のブロック構成図である。

【図12】図11の情報処理装置でプリロード命令を処理した時の処理の流れを示す図である。

【図13】本発明の第八の実施例のブロック構成図である。

【図14】本発明の第九の実施例のブロック構成図である。

【図15】本発明の第十の実施例のブロック構成図である。

【図16】図1の情報処理装置で処理する命令を説明する図である。

【図17】図16に示した命令を実行した時のキューレジスタの状態の変化を説明する図である。

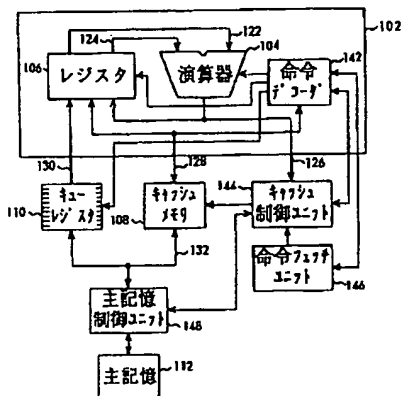
【図18】キューデータ数読み出し命令を使用したキューレジスタの退避回復処理を説明する図である。

【符号の説明】

102…命令処理ユニット、106…レジスタ、108…キャッシュメモリ、110、510、610、711、712、1012、1110、1314、1408…キューレジスタ、112、1120、1320、1430…主記憶、506…演算器、522、626、628、1034、1342、1344…データバス、904…整数レジスタ、908…浮動小数点レジスタ、912…整数用キューレジスタ、914…浮動小数点用キューレジスタ、1108…順序保証回路、1308…キューレジスタの最上部の位置を記憶するポインタ、1310…キューレジスタの最後尾の位置を記憶するポインタ、1312、1412…書き込み制御回路、1420～1423…データバッファ、1410…バンク順序記憶回路、1414…選択回路、1508、1510…キューレジスタの一部を構成するレジスタ、1512…順序保証回路。

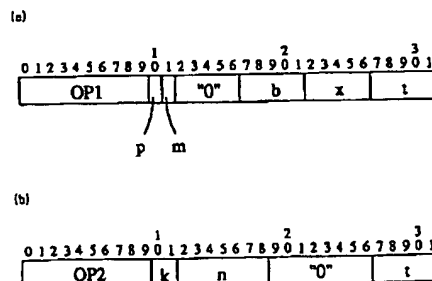
【図1】

図 1



【図2】

図 2



【図3】

図 3

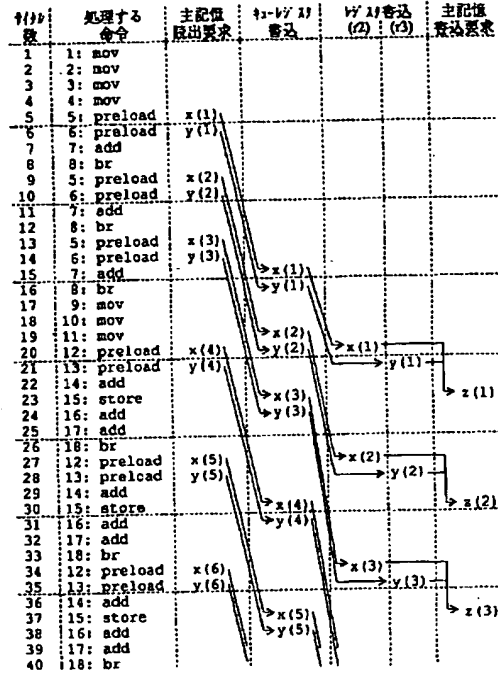
```

1:      mov     adx-1,r5
2:      mov     ady-1,r6
3:      mov     1,r0
4:      mov     3,r8
5: L0:    preload r5,r0,Q
6:      preload r6,r0,Q
7:      add     r0,1,r0
8:      br      r0,r8,<=,L0
9:      mov     adx-1,r7
10:     mov     1,r1
11:     mov     100,r8
12: L1:    preload r5,r0,Q,r2
13:     preload r6,r0,Q,r3
14:     add     r2,r3,r4
15:     store   r4,r7,r1
16:     add     r0,1,r0
17:     add     r1,1,r1
18:     br      r0,r8,<=,L1
19: L2:    preload Q,r2
20:     preload Q,r3
21:     add     r2,r3,r4
22:     store   r4,r7,r1
23:     add     r1,1,r1
24:     branch  r1,r8,<=,L2

```

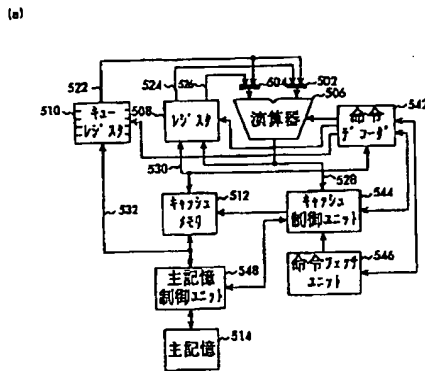
【図4】

図 4



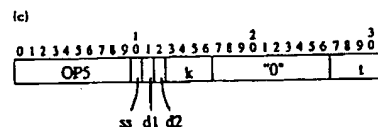
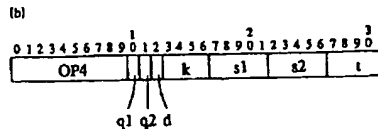
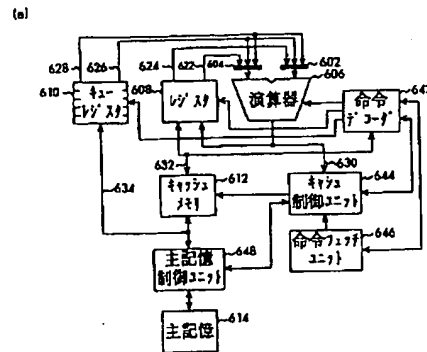
【図5】

図 5



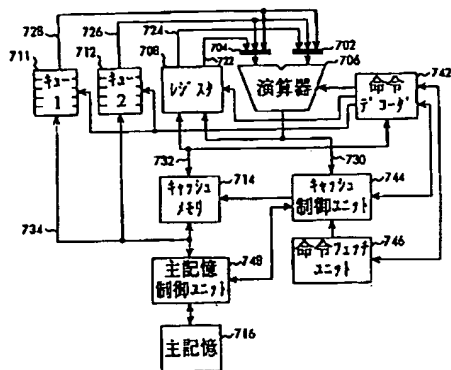
【図6】

図 6



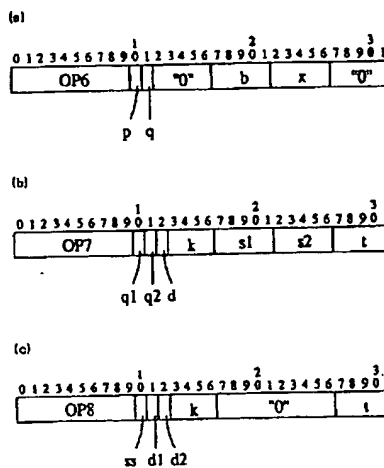
【図7】

図 7



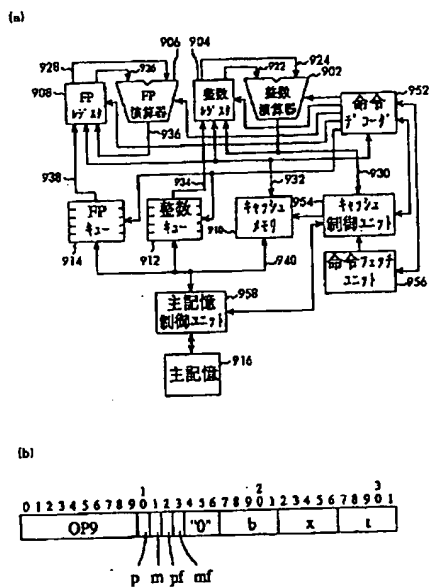
【図8】

図 8



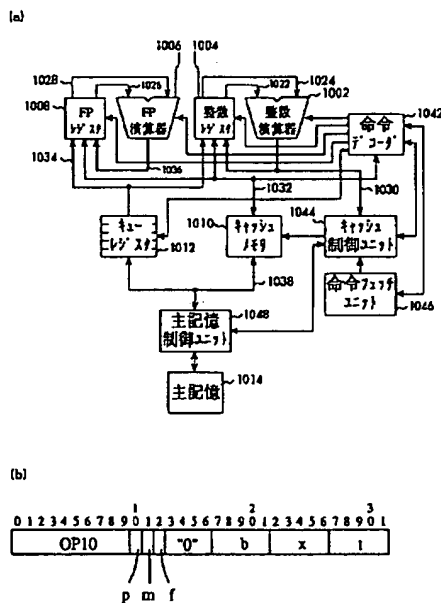
【図9】

図 9



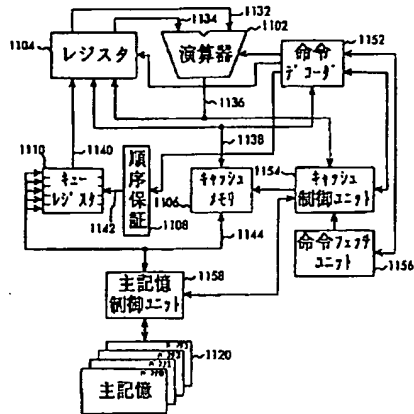
【図10】

図 10



【図11】

図 11



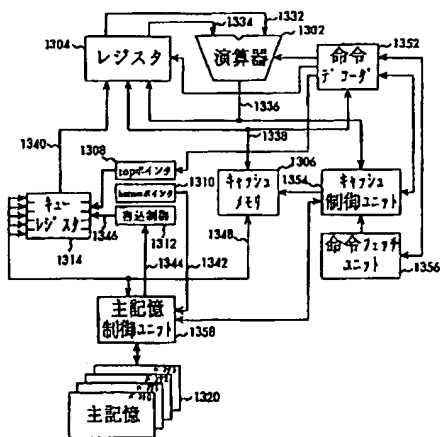
【図12】

図 12

処理する 命令 数	処理する 命令	主記憶 読出要求	メモリ 番号	キューレジスタ 番地
1	preload(a)	(a)	0	
2	preload(b)	(b)	0	
3	preload(c)	(c)	1	
4	preload(d)	(d)	1	
5	preload(e)	(e)	2	
6	preload(f)	(f)	2	
7	preload(g)	(g)	3	
8	preload(h)	(h)	3	
9				
10				
11				(d)
12				(c)
13				(e)
14				(b)
15				(g)
16				(d)
17				(f)
18				(h)
19				(a)
20				(b)
21				(e)

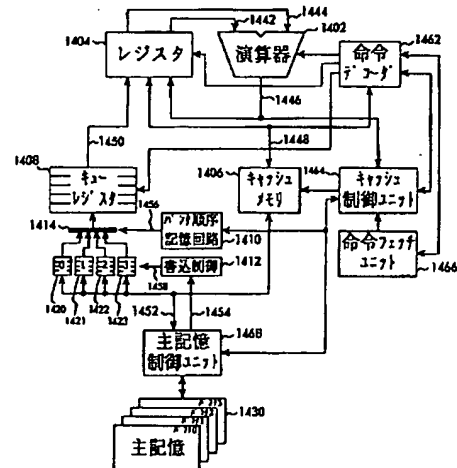
【図13】

図 13



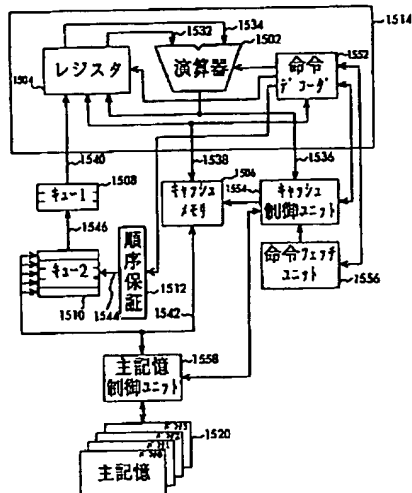
【図14】

図 14



【图 16】

16



項番	分 類	ニモニック	動 作
1	演算命令	add r, c, i	レジスタ <i>r</i> と定数値 <i>c</i> を加算し、結果をレジスタに書き込む。
2	演算命令	add r0, r1, i	レジスタ <i>r0</i> とレジスタ <i>r1</i> を加算し、結果をレジスタに書き込む。
3	ロード命令	load b, s, i	レジスタ <i>b</i> とレジスタ <i>s</i> を加算したアドレスのデータをレジスタ <i>i</i> に書き込む。
4	ストア命令	store s, b, x	レジスタ <i>b</i> とレジスタ <i>s</i> を加算したアドレスにレジスタ <i>x</i> のデータを書き込む。
5	代入命令	mov c, i	定数値をレジスタに書き込む。
6	条件分枝命令	br r0, r1, c, L	レジスタ <i>r0</i> とレジスタ <i>r1</i> を比較し、 <i>r0</i> が <i>i</i> 以下であった場合のみで示されるアドレスに分枝する。
7	プリロード命令	preload b, s, Q	レジスタ <i>b</i> とレジスタ <i>s</i> を加算したアドレスのデータをキューレジスタの最後尾に書き込む。
8	キュー転送命令	preload Q, i	キューレジスタの最上部のデータをレジスタに書き込む。
9	複合プリロード命令	preload b, s, Q, i	レジスタ <i>b</i> とレジスタ <i>s</i> を加算したアドレスのデータをキューレジスタの最後尾に書き込み、同時にキューレジスタの最上部のデータをレジスタに書き込む。
10	キュー削除命令	qdel c	定数で指定された数のデータをキューレジスタから削除する。
11	キューデータ数読み出し命令	qnum i	キューレジスタから読み出し可能なデータの数をレジスタに書き込む。

【图 18】

18

Figure 1 illustrates the scheduling of instructions in a 4-processor system. The timeline shows four processors (P0, P1, P2, P3) and their assigned instructions over time.

Timeline:

- Time 0:** P0 starts with instruction 1 (JMP 501).
- Time 1:** P1 starts with instruction 1 (JMP 501).
- Time 2:** P2 starts with instruction 1 (JMP 501).
- Time 3:** P3 starts with instruction 1 (JMP 501).
- Time 4:** P0 starts with instruction 2 (JMP 502).
- Time 5:** P1 starts with instruction 2 (JMP 502).
- Time 6:** P2 starts with instruction 2 (JMP 502).
- Time 7:** P3 starts with instruction 2 (JMP 502).
- Time 8:** P0 starts with instruction 3 (JMP 501).
- Time 9:** P1 starts with instruction 3 (JMP 501).
- Time 10:** P2 starts with instruction 3 (JMP 501).
- Time 11:** P3 starts with instruction 3 (JMP 501).
- Time 12:** P0 starts with instruction 4 (JMP 502).
- Time 13:** P1 starts with instruction 4 (JMP 502).
- Time 14:** P2 starts with instruction 4 (JMP 502).
- Time 15:** P3 starts with instruction 4 (JMP 502).

Callout Box 1 (Time 11):

```

1:  qnum  r3
2:  mov   add,r1
3:  mov   0,r2
4:  store  r3,r1,r2
5:  br     r3,r2,<=,L1
6:  add    r2,r1,r2
7:L0:  preload Q,r0
8:  store  r0,r1,r2
9:  add    r2,r1,r2
10:  br    r2,r3,<=,L0
11:L1:

```

Callout Box 2 (Time 13):

```

1:  mov   add,r1
2:  mov   0,r2
3:  load   r1,r2,r3
4:  br     r3,r2,<=,L1
5:  add    r2,r1,r2
6:L0:  preload r1,r2,Q
7:  add    r2,r1,r2
8:  br     r2,r3,<=,L0
9:L1:

```

Processor States:

- Time 11:** P0: 501, P1: 501, P2: 501, P3: 501.
- Time 13:** P0: 502, P1: 502, P2: 502, P3: 502.

(b)

実行命令	初期状態	命令1	命令2	命令3	命令4																									
実行後の メモリの 状態	B → <table> <tr><td>空</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> </table>	空	空	空	空	空	B → <table> <tr><td>1-1</td></tr> <tr><td>1-2</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> </table>	1-1	1-2	空	空	空	B → <table> <tr><td>1-1</td></tr> <tr><td>1-2</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> </table>	1-1	1-2	空	空	空	B → <table> <tr><td>1-1</td></tr> <tr><td>1-2</td></tr> <tr><td>1-3</td></tr> <tr><td>空</td></tr> <tr><td>空</td></tr> </table>	1-1	1-2	1-3	空	空	B → <table> <tr><td>1-1</td></tr> <tr><td>1-2</td></tr> <tr><td>1-3</td></tr> <tr><td>1-4</td></tr> <tr><td>空</td></tr> </table>	1-1	1-2	1-3	1-4	空
空																														
空																														
空																														
空																														
空																														
1-1																														
1-2																														
空																														
空																														
空																														
1-1																														
1-2																														
空																														
空																														
空																														
1-1																														
1-2																														
1-3																														
空																														
空																														
1-1																														
1-2																														
1-3																														
1-4																														
空																														
メモリ番地																														

実行命令	命令5	命令6	命令7	命令8
実行後の ワーレジ の状態	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $\overline{r-12}$ $\overline{r-13}$ $\overline{r-14}$ \overline{a} \overline{b} </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $\overline{r-12}$ $\overline{r-13}$ $\overline{r-14}$ \overline{a} \overline{b} </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $\overline{r-14}$ \overline{a} \overline{b} \overline{a} \overline{b} </div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $\overline{r-13}$ \overline{a} \overline{b} \overline{a} \overline{b} </div>
レジスタ書込	$\overline{r-12} \rightarrow \overline{r-1}$	$\overline{r-3}$		$\overline{r-12} \rightarrow \overline{r-14}$

(21)

特開平7-295882

フロントページの続き

(72)発明者 山縣 良
神奈川県秦野市堀山下1番地 株式会社日
立製作所汎用コンピュータ事業部内